Enhancing the performance of existing urban traffic light control through extremum-seeking

Ronny Kutadinata a,*, Will Moase a, Chris Manzie a, Lele Zhang b, Tim Garoni b

a Department of Mechanical Engineering, The University of Melbourne, Parkville, Victoria 3010, Australia
b School of Mathematical Sciences, Monash University, Clayton, Victoria 3800, Australia

ABSTRACT

Urban traffic light controllers are responsible for maintaining good performance within the transport network. Most existing and proposed controllers have design parameters that require some degree of tuning, with the sensitivity of the performance measure to the parameter often high. To date, tuning has been largely treated as a manual calibration exercise but ignores the effects of changes in traffic condition, such as demand profile evolution due to urban population growth. To address this potential shortcoming, we seek to use a newly developed extremum-seeker to calibrate the parameters of existing urban traffic light controllers in real-time such that a certain performance measure is optimised. The results are demonstrated for three categories of traffic controllers on a microscopic urban traffic simulation. It is demonstrated that the extremum-seeking scheme is able to seek the optimal parameters, with respect to a certain performance measure, for each of these traffic light controllers in an urban, uni-modal traffic environment.

1. Introduction

There have been major developments in intelligent control of urban traffic lights over the last few decades. These advanced control strategies have been implemented either in a traffic simulator or in real life with varying degrees of success reported under different operating conditions. An ideal urban traffic light control strategy should be able to cope with all traffic conditions, namely undersaturated, saturated or oversaturated. Furthermore, it should be adaptive to real-time traffic conditions and robust against disturbances, such as traffic demand fluctuations, accidents, road works or special events. Therefore, recent development of urban traffic light control follows the trend of using some real-time measurements as feedback in a closed-loop (online) control strategy to correct for measured changes in the traffic.

A closed-loop (online) traffic light control can be classified as model based or non-model based. A model based controller uses a prediction model of the traffic system to find an appropriate traffic light input. Two common examples of model based strategies are: rolling horizon model predictive control (Henry et al., 1983; Gartner, 1983; Mirchandani and Head, 2001; Tettamanti and Varga, 2010; Lin et al., 2012; Aboudolas et al., 2010) and linear quadratic regulator/integrator (LQR/I) (Diataki et al., 2002; Aboudolas and Geroliminis, 2013). A rolling horizon model predictive control (MPC) coupled with the use of a high-fidelity model results in a computational-heavy controller. Hence, the use of MPC is limited to controlling
one intersection or a few at best (Henry et al., 1983; Henry, 1989; Gartner, 1983; Boilot et al., 1992; Mirchandani and Head, 2001). To address this computational issue, linear models are utilised (Diakaki et al., 2000; Dinopoulou et al., 2000; Bielefeldt et al., 2001; Diakaki et al., 2002, 2003; Dinopoulou et al., 2006; Papageorgiou et al., 2006; Lin et al., 2009; Aboudolas et al., 2010; Lin et al., 2010, 2011), which typically use “averaged” parameters, and the performance of the controller is dependant on how well these parameters are approximated.

When the concern is emphasised on the congestion in the whole network rather than looking at each individual street, it is also possible to model the whole network as a group of smaller regions, where each region is modelled by an aggregate dynamical equation that uses the concept of the Macroscopic Fundamental Diagram (MFD). A MFD describes the relationship between vehicle accumulation and the flow within a traffic network. Among other literature on MFD (Wardrop, 1968; Godfrey, 1969; Herman and Prigogine, 1979; Mahmassani et al., 1987; Ardekani and Herman, 1987; Daganzo, 2007), analysis has revealed that the shape of MFD is concave (Daganzo and Geroliminis, 2008) and it has been verified by experimental results (Geroliminis and Daganzo, 2008). Further studies for different scenarios of urban traffic networks also show that the concave nature of the MFD always presents (Aboudolas et al., 2010; Zhang et al., 2013; Geroliminis et al., 2014). Therefore, this is exploited in a perimeter control scheme (Aboudolas and Geroliminis, 2013; Keyvan-Ekbatani et al., 2012, 2013; Haddad and Shraiber, 2014; Keyvan-Ekbatani et al., 2015; Ramezani et al., 2015), which essentially is a gating strategy that limits the incoming traffic flow such that the vehicle accumulation in each region stays close to its critical value that leads to optimal throughput/flow of the network. Perimeter control schemes use vehicle conservation models, where the number of vehicles that complete their trips (throughput) is predicted by using the MFD and the number of vehicles that enter the network is restricted by the perimeter control. The goal of perimeter control is to regulate the vehicle accumulation within the network to a set-point. However, since the MFD of the network might vary for different traffic conditions (Zhang et al., 2013), the performance of the controller largely depends on the set-point that is manually selected.

On the other hand, an online non-model based approach establishes “rules” to be followed. Some examples are: SCATS (Lowrie, 1982) and self organising traffic lights (SOTL) (Gershenson, 2005; de Gier et al., 2011). Similar to the other traffic light controls, an online non-model based controller’s performance depends on its tuning parameters, such as offset, or maximum cycle time in the case of SCATS (Lowrie, 1982), and threshold value in the case of SOTL (de Gier et al., 2011; Zhang et al., 2013). Therefore, regardless of whether the controller is model based or model-free, the use of a parameter adaptation technique as an augmentation to these controllers is potentially beneficial to their performances. In fact, there are some previous works that have investigated parameter adaptation of traffic light controllers.

The adaptation techniques that were used in existing literature for adapting the parameters of traffic light controllers are SPSA and its modification (referred to as the AFT technique) (Spall, 1992, 2003; Maryak and Chin, 2008; Spall and Chin, 1994, 1997; Chin et al., 1999; Kosmatopoulos et al., 2008a,b; Kosmatopoulos, 2009; Kouvelas et al., 2011). These works incorporate the parameter adaptation scheme as a higher level control in a hierarchical manner, where the SPSA is performing a higher-level control on the existing traffic light controller by providing parameters to optimise its performance. However, there has not been any rigorous stability proof of either SPSA and AFT applied to systems with dynamics; and extending the existing stability result to include dynamical systems might be a challenge due to the discrete nature of these techniques. Therefore, it is proposed to use an extremum-seeking technique for the parameter adaptation, since its stability has been rigorously guaranteed for application on dynamical systems.

Extremum-seeking (ES) is a non-model based steady-state optimisation scheme for dynamical plants. An ES controller regulates the input of a dynamical plant to the value that optimises the steady-state output (i.e. cost or performance measure) of the plant, without requiring knowledge of the underlying dynamics. In order to achieve this, ES requires several components, namely the dither signals, the gradient estimator, and the optimiser operating in progressively slower time-scales. There are many ways to realise these components (Spall, 1992; Krstić and Wang, 2000; Chichka et al., 2006; Srinivasan, 2007; Moase et al., 2010; Liu and Krstić, 2010), the simplest of which is outlined by Tan et al. (2006). While there have been a multitude of ES applications demonstrated over the past decade, the vast majority are on low dimensional (usually single-input single-output) systems.

In the case of controlling the traffic lights of an urban network, the number of parameters typically grows with network size, which along with communication considerations, motivates the use of a decentralised approach of ES. A type of a multi-input multi-output (MIMO) ES scheme that can solve this kind of problem is the Nash equilibrium seeking (NES) (Stanković et al., 2010; Frihauf et al., 2011; Kutadinata et al., 2015). Kutadinata et al. (2015, in preparation) demonstrate the stability of a NES scheme acting on a class of hybrid systems, thereby bridging the gap introduced by previous application of SPSA and AFT to a similar problem.

This paper extends the study of Kutadinata et al. (2014) and investigates the benefit of using a NES scheme for fine-tuning the parameters of several existing urban traffic light control strategies, as illustrated in Fig. 1. This paper investigates the calibration of the parameters of three existing urban traffic light controllers taken from the closed-loop model-free and model-based families of traffic controllers: SCATS’s internal and orthogonal offsets; SOTL’s threshold value; and the density set-point of a perimeter control. The remainder of this paper is organised as follows. Section 2 outlines the NES scheme used, and is followed by the description of the simulation set up in Section 3. The rest of the paper outlines the simulation results of calibrating perimeter control (Section 4), SCATS (Section 5), and SOTL (Section 6).
2. Overview of the Nash equilibrium seeking scheme

In this section, the proposed extremum-seeking scheme is outlined and some of its features are explained. Specifically, Fig. 1 is to be explained in more detail to clarify the control objective and the process of the whole closed-loop system.

2.1. Control objective in a non-cooperative game setting

There are three parts of Fig. 1 that are of main concern in this section: the plant’s input $u$, the plant’s output $y$, and the NES scheme (the bottom dashed box). From the perspective of the NES scheme, the combination of the traffic network and the traffic light controller is a single plant. Thus, the plant’s input $u$ is the parameter vector of the traffic light controller, and the output $y$ is the performance measure.

The parameter selection affects the performance of the traffic light controller, which is measured as the plant’s output $y$. The plant’s output $y$ can represent various performance measures, such as (but not limited to) average speed or flow. Given the dimension of $u$ for the full network, it is desirable to decentralise the NES scheme and as a result, the output is not a network-wide performance measure, but rather a group of “local” performance measures (i.e. $y$ is a vector), each only considers a smaller part of the network. Then, each parameter is associated with a “local” performance measure/cost function/output (these three terms are used interchangeably throughout this paper) and it is desired to find the value that optimises its associated local output in the steady-state (note that ES is a steady-state optimisation).

In order to understand the decentralisation process, consider changing the setting of the traffic lights at an intersection. It is reasonable to expect the effect of a given set of traffic lights to decay with “distance”. Note that this decay is not solely determined by physical distance, as there are potentially other factors that contribute to the decay, such as network topography and origin–destination profile (which determines the direction of the majority of the traffic flows). Then, consider changing a particular parameter which affects the traffic light setting of a group of intersections. Possible examples are changing a threshold value in SOTL only affects one intersection whereas changing an offset in SCATS affects several intersections. As a result, each parameter is expected to affect the traffic condition only locally. Then, associating only a local performance measure (instead of the network-wide performance measure) for each parameter is justifiable in this case.

However, if a local performance measure associated with a given parameter is still affected by the other parameters (albeit not significantly), it is possible that optimising one of the local outputs results in the deterioration of the other local outputs. For instance, increasing the flow out of an intersection might create congestion at downstream intersections, which in turn decreases the outflows of those downstream intersections. This eventually leads to a non-cooperative game being played and each parameter acts as a player. Since each “player/agent” is only optimising the local cost selfishly, this leads to the Nash equilibrium. Specifically, let $u \in \mathbb{R}^{N_u}$ and $u_j$ be the $j$th input/parameter and $J_j(u)$ be the associated steady-state local output of $u_j$ (i.e. the output associated with the $j$th parameter $y_j \rightarrow J_j(u)$ as $t \rightarrow \infty$). Then, the Nash equilibrium, $u^*$, is defined as a point which satisfies

$$\frac{\partial J_j}{\partial u_j}(u^*) = 0, \quad \frac{\partial^2 J_j}{\partial u_j^2}(u^*) < 0,$$

for all $j \in \{1, \ldots, N_u\}$. Having defined the Nash equilibrium, the next subsection describes a NES scheme.

Note that although the local performance measure of each player/agent may not be explicitly related (for instance, each agent can consider an independent subregion of the network), in the case where the interconnection between the local performance measures of the agents is inherent, a multi-agent decentralisation (such as the one proposed in this subsection)
inevitably creates a non-cooperative game problem. Therefore, the following NES scheme is most suitable to address such optimisation as it directly considers the competition among these agents.

### 2.2. Nash equilibrium seeking with linear filters

The NES scheme consists of two parts. Firstly, the output associated with each input $y_j \in \mathbb{R}$ is processed through a group of $N_u$ linear filters. Each filter is as follows:

\[
\begin{align*}
\dot{\xi}_j &= \omega A_j \xi_j + \omega B_j y_j, \\
\dot{y}_j &= C_j \xi_j,
\end{align*}
\]

where $\xi_j \in \mathbb{R}^N$ is the filter’s state, $A_j \in \mathbb{R}^{N_x-N_t}$ is chosen to be Hurwitz, $B_j \in \mathbb{R}^{N_t}$, $C_j \in \mathbb{R}^{1 \times N_x}$, $\dot{y}_j \in \mathbb{R}$ is the filter’s output, and $\omega > 0$ is a parameter used to introduce time-scale separation between the plant and the NES scheme, i.e. by making $\omega$ sufficiently small, the NES scheme is made to act slowly relative to the plant dynamics. The linear filters used in this work are a group of identical band-pass filters,

\[
A_j = \begin{bmatrix} -\omega_l & 0 \\ -\omega_h & -\omega_l \end{bmatrix}, \quad B_j = \begin{bmatrix} \omega_h \\ \omega_l \end{bmatrix}, \quad C_j = [0 \ 1],
\]

with cut-off frequencies at $\omega_h$ (high-pass filter) and $\omega_l$ (low-pass filter). These band-pass filters “extract” the components of the output signals which lie within a range of frequencies corresponding to those of the dither signals.

The second part of the NES scheme involves correlating the dither with the plant output,

\[
\hat{u}_i = k \omega \sin(\omega \omega_l t - \phi_i) \hat{y}_j,
\]

where $\hat{u}_i$ is the nominal value of the $j$th input, $\omega_j \in \Omega_{\omega_{\text{max}}}$ for all $j \in \Sigma$ and some $\omega_{\text{min}} > 0$ where $\Omega$ is the set of rational numbers, and $\phi_i \in [0, 2\pi)$. The resulting effect is to adapt $\hat{u}_i$ at a rate that is, on average, proportional to $\partial J_j / \partial u_j$ (recall that $J_j$ is the associated steady-state local output of $u_i$, i.e. $y_j \rightarrow J_j(u)$ as $t \rightarrow \infty$). Hence, the complete NES scheme can be written:

\[
\begin{align*}
\dot{\xi} &= \omega A \xi + \omega B y, \\
\dot{u} &= -k \omega \text{diag}(s(\omega \omega_l t - \phi)) C \xi, \\
u &= \hat{u} + a s(\omega \omega_l 0),
\end{align*}
\]

where

- $\xi \in \mathbb{R}^N$ and $N_t = \sum_{j=1}^{N_u} N_j$;
- $\hat{u} \in \mathbb{R}^{N_u}$;
- $\text{diag}(\cdot)$ is a diagonal matrix where the diagonal elements are given by its argument;
- $A := \oplus_{j=1}^{N_u} A_j, \ B := \oplus_{j=1}^{N_u} B_j, \ C := \oplus_{j=1}^{N_u} C_j$, where $\oplus$ is the direct sum;
- $s(\omega \omega_l t - \phi)$ is a dither/perturbation vector whose $j$th element is given by $\sin(\omega \omega_l t - \phi));
- k \omega$ is the dither frequency of input $j$;
- $a \in \mathbb{R}_{>0}$ is the dither amplitude;
- and $k \in \mathbb{R}_{>0}$ is the adaptation gain.

The role of $\phi$ is to partially compensate for the phase lag introduced by the plant dynamics and filters.

### 2.3. Dither re-use

Considering the decentralisation characteristic of the problem discussed in Section 2.1, it is proposed that this be exploited to simplify the tuning process of the NES scheme, which provides some practical benefits (Kutadinata et al., 2015). In particular, it is possible to use fewer unique dither frequencies within the NES scheme, whereas a typical multi-input ES scheme requires the dither frequency of each input to be unique. It has been shown that two inputs who have sufficiently weak effects on each other’s cost may use the same dither frequency. Precise mathematical explanation of the dither re-use is outlined by Kutadinata et al. (2015).

### 2.4. Dealing with day-to-day traffic fluctuations

In some situations, it is desirable to have multiple parameter set-points in a traffic controller during a day, reflecting the presence of varying load demands. For instance, a day can be divided into three periods each with its own traffic demand, namely evening, peak, and off-peak, which occurs during different time of the day. In order to accommodate this situation, several parallel NES controllers are set-up, where each NES controller is activated for each period. Therefore, the NES scheme remembers its state from the previous cycle and can continue the optimisation without having to start over. In particular, the
NES resets the value of \( \xi, \bar{u} \), and the dithers' phases at the beginning of a period to the final values the last time that period was entered.

3. Simulation description

3.1. Traffic network

In this paper, the considered square network consists of 16 intersections (nodes), as shown in Fig. 2. Each road (link) consists of two lanes and one right-turning lane,\(^1\) as shown in Fig. 3a. The lengths of the road and the right-turning lane are 300 m and 45 m respectively. This is indicative of the length of roads in typical Australian city centres. In this paper, a cellular automata model, which is described in detail by de Gier et al. (2011), is used in the simulations (Fig. 3b). Furthermore, each intersection uses these four phases: an east/west phase, a north/south turning phase, a north/south phase, an east/west turning phase (Fig. 4). Note that in this work, the simulated traffic is uni-modal, i.e. all vehicles are identical. In addition, to simplify the terminology, each road is categorised as either a boundary road (i.e. a road that has an entry/exit point from/to the unsimulated region) or a bulk road (otherwise). For the remainder of this paper, the terms intersection and node are used interchangeably, as are road and link.

Driver behaviour in the simulation is modelled via a series of stochastic processes. Firstly, at each time step, there is a probability of \( \alpha \) that a vehicle enters the network through each inbound boundary road. To simulate congestion outside the network, a vehicle exits the network with probability \( \beta \) when it reaches the end of an outbound boundary road. \( \alpha \) and \( \beta \) can be given a subscript that indicates its direction, as shown in Fig. 2. In addition, there are internal traffic sources and sinks located near the middle of each lane of each bulk road, representing, for example, car-parks. The probabilities of a vehicle entering and exiting the network via the internal sources and sinks are \( \gamma \) and \( \delta \) respectively. Various traffic conditions can be simulated by varying \( \alpha, \beta, \gamma, \) and \( \delta \). Furthermore, a predetermined turning probability, both left and right, enables each vehicle, at each node, to make a random decision regarding its direction. In this study, the turning probability is fixed at 10% in each direction. Table 1 summarises the traffic condition parameters that are varied in this study. The traffic conditions used for each experiment is different and specifically selected to emphasise the effect of the varied parameters.

3.2. Traffic light controllers augmented with NES scheme

The implementation of the NES scheme to augment a traffic light control strategy is as follows. Given any arbitrary signal control strategy with tunable parameters, each parameter to be tuned becomes an agent. Next, the performance measure of each agent is determined, which should be designed to reflect the aim of the optimisation. In addition, the performance measure design also determines the interconnection between agents. The performance measure of an agent can accommodate only some parts of the whole network, and it can overlap with other agents’ considered regions. Afterwards, the variable gains of the NES scheme is determined, including the dither amplitude \( a \), the adaptation gain \( k \), and the dither frequencies.

The following implementation steps is applicable when dealing with many kinds of traffic light controller of various categories.

In this paper, three traffic light controllers are studied: SCATS, SOTL and perimeter control. SCATS (Lowrie, 1982; Zhang et al., 2013) is chosen to demonstrate the usefulness of the proposed NES scheme when applied to one of the most widely used traffic light controllers. Meanwhile, the chosen version of SOTL (Zhang et al., 2013) and perimeter controller (Aboudolas and Geroliminis, 2013; Ampountolas et al., 2014) are two of the latest urban traffic light controllers proposed in the non-model and model based categories, respectively. Thus, these two are chosen to emphasise the versatility of the proposed NES scheme to future implementations. Each of the three controllers has parameters that are selected adaptively using the NES scheme as summarised in Table 2.

3.2.1. Perimeter control

The first traffic light controller considered is perimeter control. The development of this strategy largely depends on the existence of a Macroscopic Fundamental Diagram (MFD) within a certain homogeneous region/network. The MFD demonstrates traffic flow decreases away from a unique critical density. This controller is essentially a gating strategy aimed to regulate the density inside the network to a certain set point by restricting the incoming traffic flow into the network (the perimeter control used in this paper is based on the work by Aboudolas and Geroliminis (2013), refer to Appendix A for a more detailed description). The NES scheme is then used to fine-tune the selection of the density set-point such that the performance measure is optimised. In this case, the whole network is considered as a region that involves only one input and output, in which case the NES scheme degenerates and becomes a single-input, single-output ES scheme. Since there is only one input (representing one agent), the achieved extremum is a regular maximum point and not the Nash equilibrium. In addition, only the bulk roads are considered to be parts of the network that are protected by the gating strategy, and the boundary roads are simply seen as buffers to couple the bulk network to the unsimulated region.

\(^1\) Vehicles drive on the left in this model.
Fig. 2. The traffic network used in the simulation. Each intersection is numbered as shown in the figure. The subscripts of $\alpha$ and $\beta$ indicate the directions of the denoted traffic flow.

Fig. 3. The illustration of the traffic model used.

Fig. 4. The phases at each intersection simulation: east/west phase (phase 1), north/south turning phase (phase 2), north/south phase (phase 3), east/west turning phase (phase 4). A dashed line indicates a traffic flow that has to give way to other flows, whereas a solid line indicates a flow that has the right of way.

Table 1
Varied traffic condition parameters.

<table>
<thead>
<tr>
<th></th>
<th>Entry probability</th>
<th>Exit probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boundary link</td>
<td>$\alpha$</td>
<td>$\beta$</td>
</tr>
<tr>
<td>Bulk link</td>
<td>$\gamma$</td>
<td>$\delta$</td>
</tr>
</tbody>
</table>
Overview of the considered scenarios.

<table>
<thead>
<tr>
<th>Traffic light controller</th>
<th>Type of controller</th>
<th>Calibrated parameters</th>
<th>No. of parameters</th>
<th>Performance measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perimeter</td>
<td>Online, model</td>
<td>Density set-point</td>
<td>1</td>
<td>Mixed avg. speed &amp; flow</td>
</tr>
<tr>
<td>SCATS</td>
<td>Online, non-model</td>
<td>Internal, orthogonal offsets</td>
<td>7</td>
<td>Avg. speed</td>
</tr>
<tr>
<td>SOTL</td>
<td>Online, non-model</td>
<td>Threshold values</td>
<td>16</td>
<td>Avg. speed</td>
</tr>
</tbody>
</table>

Although the aim of the perimeter control is to achieve maximum flow in the network, it is common for the desired density set point to be just below the critical density (Aboudolas and Geroliminis, 2013; Ampountolas et al., 2014; Haddad and Shraiber, 2014). In order to achieve this, it is proposed that a performance measure that is a linear combination of the average speed and flow be used. Specifically, let \(v_i(t)\) be the speed of vehicle \(i\) at time \(t\) and \(N_j(t)\) be the set of the vehicles that are on the roads that are considered by parameter \(j\), at time \(t\). Also, to ease presentation and to reduce the fluctuations in the performance measure, it is subjected to a moving average filter with averaging period of \(T_{av} = 5\) min. Then, the average speed associated with parameter \(j\) is defined as:

\[
y_{speed_j}(t) = \frac{1}{T_{av}} \int_{t-T_{av}}^{t} \left( \frac{1}{|N_j(s)|} \sum_{i \in N_j(s)} v_i(s) \right) ds.
\]  

Furthermore, let \(f_j(t)\) be the impulse signal from the sensor on lane \(l\) (located 45 m downstream of the upstream intersection) at time \(t\), and \(L_j\) be the set of lanes that are considered by parameter \(j\). The average flow (per lane) associated with parameter \(j\), \(y_{flow_j}(t)\), is calculated in a similar manner to the average speed in (5), as follows

\[
y_{flow_j}(t) = \frac{1}{T_{av}} \int_{t-T_{av}}^{t} \left( \frac{1}{|L_j(s)|} \sum_{i \in L_j(s)} f_i(s) \right) ds.
\]

Then, the mixed network performance measure is as follows

\[
y_{mixed}(t) = y_{flow_j}(t) + w_s y_{speed_j}(t),
\]

where \(y_{flow_j}\) and \(y_{speed_j}\) are the aforementioned average speed and flow associated with parameter \(j\), and \(w_s\) has units veh/m.

In addition, the performance measure at an intersection refers to the averaged performance measure of the links that are approaching the intersection.

### 3.2.2. SCATS

The version of SCATS in this paper adaptively controls two key traffic light settings: cycle length and split time, at each intersection. A comprehensive description is provided in Zhang et al. (2013). On top of that, the NES scheme adjusts the internal and orthogonal offsets in the network.

The internal offset determines the delay of the start of the east/west phase of a node relative to that of the node directly to its west. Since each link is identical, the same internal offset is used along a series of nodes (from east to west). This provides green waves for the east bound traffic and creates four one-dimensional subsystems in the network (Fig. 5) and four internal offsets, one for each subsystem. Furthermore, the orthogonal offset represents the delay of the start of the east/west phase of a node that is located at the east edge of the network relative to that of the node directly at its south. Lastly, a master node indicates the “starting point” of the green wave within a subsystem and the super master node is the starting point of the “orthogonal green wave” along the master nodes (otherwise a node is referred to as a slave). Hence, there are four internal offsets and three orthogonal offsets. Since the main aim of using offsets is to create a green wave and reduce the chance that vehicles are stopped by red light, it is sensible to use the average speed as the performance measure of SCATS.

The realisation of the NES scheme is more involved since SCATS has more parameters to be tuned. There will be seven NES agents corresponding to the seven offsets to be tuned. By considering the subsystems and offsets arrangements, it is reasonable that each internal-offset-agent considers only its own subsystem. Specifically, the internal-offset-agent aims to optimise the performance measure at the intersections that are included in the subsystem within which the internal offset is applied. Furthermore, since each orthogonal offset is linking two adjacent subsystems, it is sensible to assign those two subsystems as the objective of the corresponding agents. Specifically, the aim of each orthogonal-offset-agent is to optimise the performance measure at the intersections that are included in the two subsystems that are being linked by the orthogonal offset. The effect of the offsets on the performance of the intersections are discussed in Section 5.1.

### 3.2.3. Self-organising traffic light

The self-organising traffic light scheme used in this work is explained in detail by de Gier et al. (2011) and Zhang et al. (2013). SOTL is a priority based control that measures in real-time the value of the threshold function of a phase, which represents the “busyness” of a phase, at each intersection. When the measure of the threshold function exceeds a threshold value, it becomes a candidate for the next active phase. The selection of the threshold value changes how quickly the phase
switches, and hence affects the performance of SOTL (de Gier et al., 2011). Then, the NES scheme is used to regulate the threshold values to the Nash equilibrium. Lastly, since SOTL in its essence is an isolated intersection control, it is mainly concerned with reducing the delay of vehicles attempting to cross an intersection. Thus, the measure of average speed used for SCATS is also suitable for SOTL.

Since the SOTL strategy uses one threshold value at each intersection, there are 16 agents (i.e. 16 parameters to be tuned) in this case. In addition, since this version of SOTL is essentially an isolated intersection control, the objective of each agent is to optimise the performance measure at its own intersection.

4. Simulation results for a model based strategy augmented with NES

An example of a model based strategy that is investigated is the network perimeter control proposed by Aboudolas and Geroliminis (2013). In this paper, the perimeter controller is implemented by varying \( \alpha \). Since perimeter control in practice is only applied when the traffic demand is sufficiently high, then it is assumed that at all times there are always vehicles that want to enter the network. Hence, varying \( \alpha \) is equivalent to the process of restricting the proportion of the traffic flow demand that is allowed to enter the network. Since the perimeter controller adjusts the value of \( \alpha \), the traffic condition is determined by the value of \( \beta, \delta \) and \( \gamma \). Moreover, SOTL is used as the intersection control inside the network with threshold value equal to 5. It was shown that SOTL performs well on homogenising the density on the links in a network (Zhang et al., 2013), which is essential to have a well-defined and robust MFD.

This section provides the simplest implementation of NES among the three traffic light controllers considered as there is only one parameter updated (from Table 2). There are two aspects being investigated. The steady-state input–output (SSIO) maps for several traffic conditions are first obtained. The SSIO map shows the relationship between the calibrated parameter and the average steady-state value of the performance measure, under a given time-invariant traffic condition. Note that in practice, obtaining the SSIO map is often not possible and is unnecessary other than to justify NES as an appropriate augmentation algorithm. The benefit of the proposed NES scheme is then tested on several relevant time-invariant traffic scenarios.

4.1. Steady-state input–output map

Note that as previously mentioned, the boundary roads are not considered to be parts of the network. As a result, the network aggregate observables are calculated by considering only the bulk roads. The steady-state value of the mixed performance measure at two time-invariant traffic conditions is plotted against the density set-point of the perimeter control (Fig. 6), which is the SSIO map for the NES scheme. Note that the flatness of the maps above a certain density set-point is caused by the fact that, due to the traffic conditions (i.e. \( \beta, \gamma \) and \( \delta \)) being constant, the system is unable to reach the specified density. The NES scheme cannot be guaranteed to converge to the optimum if initialised in this region. Nevertheless, the optimal density set-point is not located in that flat region and is able to be sought when the NES scheme is initialised properly. Moreover, it can be observed that the optimal density set-point can vary greatly for different traffic conditions (especially for different \( \gamma \) and \( \delta \)).

Fig. 5. The set-up of the subsystems, the master and the super master nodes\(^2\). Each shape/node represents an intersection, and the links represent the roads connecting the intersections.
Therefore, the NES scheme can potentially provide benefits to the performance of the perimeter control by fine-tuning the set point. In addition, the knowledge of the SSIO map in practice is unlikely available or is difficult to obtain. Hence, the role of the NES parameter adaptation scheme is crucial when determining the optimal set-point without knowing the SSIO map.

4.2. Time-invariant traffic demand

The NES parameters are set up as follows: dither amplitude \( a = 0.025 \); \( k = 4 \); \( \omega = 5 \times 10^{-4} \); \( \omega_H = 9 \times 10^{-5} \) rad/s; \( \omega_L = 2.5 \times 10^{-3} \) rad/s; and \( \phi = 70^\circ \).

For this section, two time-invariant traffic demand profiles are considered (shown in Table 3), which approximate two different types of peak conditions. Case 1 reflects a peak condition of a suburban city centre where only some of the vehicles finish their trips inside the network, while the other are passing through (indicated by low congestion outside the network/high \( \beta \), small sources \( \gamma \) and sinks \( \delta \)); whereas Case 2 is aimed to approximate a peak period of a CBD region where most vehicles are entering car parking spaces inside the network (medium congestion outside the network/medium \( \beta \), negligible sources, and larger sinks – twice the size of \( \delta \) in Case 1). Note that traffic conditions with large sources are not considered since in that case, perimeter control is unlikely to perform efficiently (the perimeter control would just restrict cars from entering the network at its boundaries).

The NES scheme uses \( w_s = 0.005 \) such that it emphasises maximising flow rather than speed, and the mixed performance measure should just push the optimal set point slightly lower than the critical density. It can be observed from Figs. 7 and 8 that the NES scheme is able to converge to two different equilibria for these two cases. In addition, the mixed performance measure for both cases is improved. Note that the large fluctuations of the average speed in Fig. 8 is caused by the fluctuations of the density (introduced by the set-point change in the perimeter control) around a point where the gradient of the speed vs. density graph is coincidentally steep.

Therefore, it has been demonstrated that the NES scheme is capable of seeking the optimal density set-point, improving the performance of the perimeter control. Similar results can be achieved when augmenting other model-based traffic control strategies, tuning parameters and performance measures that satisfy the general assumptions of NES.

5. Simulation results for SCATS augmented with NES

In this section, NES is used to augment SCATS. Compared to perimeter control, this task involves real-time adaptation of more parameters. Similar to Section 4, the SSIO maps of SCATS are presented and the convergence of the NES scheme when calibrating the offsets is demonstrated for several time-invariant traffic conditions. Note that the maximum cycle length used is 90 s (excluding four 3-s waiting periods between phases).

5.1. Steady-state input–output map

The aim of this section is to understand the effect of the offset (either orthogonal or internal) on the average speed of each intersection. In order to do that, there are several traffic conditions that are investigated, as shown in Table 4. Essentially, the cases considered are traffic conditions which are biased in one direction, with lower demand from all other directions. In addition, the intensity of the lower demand is varied as well. Since the phase-linking is most efficient when the vehicles are free-flowing, \( \beta = 0.99 \) and \( \gamma = \delta = 0 \) for all cases such that the vehicle platoons are not interrupted.
Since the main aim of the internal offset is to benefit the prioritised flow, i.e. the east bound flow, it is desired to find an internal offset at which the average speed of the east bound flow is maximum. The internal offset of subsystem 2 is varied, while the other internal offsets are set to 20 s (based on an approximated travel speed of 55 km/h) and the orthogonal offsets are set to 0 s. Firstly, Case 1 is considered. As shown in Fig. 9, the optimal internal offset is roughly 15 s indicating free-flowing traffic through multiple intersections, i.e. a green wave exists. In addition, because Case 1 almost does not have any incoming traffic from the other directions, the effect of the internal offset on the average speed of the flow in the other directions is insignificant.

Moreover, Fig. 10 shows the plots of the average speed from all directions at the intersections of subsystem 2. It can be observed that the shapes of the plots for Case 1 follow closely the shapes of the east bound plots in Fig. 9. When the demand from the other directions is increased (changing from Case 1 to Case 5), the plots in Fig. 10 do not exhibit optima that are similar across the nodes in the subsystem, as those of Fig. 9. This shows that the benefit of the internal offset is most obvious when the traffic is heavily biased in the east direction. In addition, the effect of the orthogonal offset on the north/south bound traffic is similar to the effect of the internal offset on the east/west bound traffic (result is not shown for brevity).

### Table 3
Perimeter control time-invariant traffic demand.

<table>
<thead>
<tr>
<th></th>
<th>$\beta$</th>
<th>$\gamma$</th>
<th>$\delta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case 1</td>
<td>0.60</td>
<td>0.05</td>
<td>0.10</td>
</tr>
<tr>
<td>Case 2</td>
<td>0.40</td>
<td>0.01</td>
<td>0.20</td>
</tr>
</tbody>
</table>

Fig. 7. (Perimeter control) The simulation result for Case 1. Note that density is calculated similar to average speed in [5].
In conclusion, with this choice of parameter combination and set-up, the offsets are beneficial only to the prioritised traffic flow and have an opposing effect on the opposite flow. As a result, when the demand of these opposing directions is approximately equal, the offset does not provide any clear overall benefit. Therefore, the usefulness of offsets is most obvious when the traffic demand is heavily biased in only non-opposing directions. For instance, the traffic condition during peak hours can most likely benefit from properly tuned offsets.

5.2. Time-invariant traffic demand

Since, in the current set-up the offsets are only clearly beneficial when the traffic demand is biased in non-opposing directions, the proposed NES scheme is tested only on a traffic demand scenario that is biased on the East and North directions. However, two different speed limits are applied to demonstrate that the NES scheme is able to locate different optima. The practice of using different speed limits during different periods of the day is common in some cities in Australia, such as in school zones where the speed limit switches between 40 and 60 km/h. However, the speed of a vehicle in the simulation is

Table 4
(SCATS) East bound biased traffic.

<table>
<thead>
<tr>
<th></th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
<th>Case 4</th>
<th>Case 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_E$</td>
<td>0.16</td>
<td>0.16</td>
<td>0.16</td>
<td>0.16</td>
<td>0.16</td>
</tr>
<tr>
<td>$x_W$</td>
<td>0.01</td>
<td>0.03</td>
<td>0.05</td>
<td>0.07</td>
<td>0.09</td>
</tr>
<tr>
<td>$x_N$</td>
<td>0.01</td>
<td>0.03</td>
<td>0.05</td>
<td>0.07</td>
<td>0.09</td>
</tr>
<tr>
<td>$x_S$</td>
<td>0.01</td>
<td>0.03</td>
<td>0.05</td>
<td>0.07</td>
<td>0.09</td>
</tr>
</tbody>
</table>

Fig. 8. (Perimeter control) The simulation result for Case 2.
expressed in cells/s (each cell is 7.5 m long), ranging from 0 to \( v_{\text{max}} \) cells/s. Due to the stochasticity in the simulation (particularly the random deceleration done by each vehicle; see Zhang et al. (2013)), the average free-flow speed that can be achieved is actually lower than \( v_{\text{max}} \). By setting the maximum vehicle speed to 2 cells/s and 3 cells/s, the average free-flow speed achieved is approximately 45 km/h and 60 km/h respectively. This closely represents the speed limit decrease of school zones. The complete traffic conditions used are shown in Table 5. Based on previous study (Kutadinata et al., 2014), these traffic conditions are approximately of medium density but the vehicle platoons are expected to still be free-flowing in order to focus the study only on the effect of offsets.

Based on the observations in the previous subsection, the “cost/output” associated with each offset is set up as follows by using the subsystems shown previously in Fig. 5. Orthogonal offset 1 considers the average speed of subsystems 1 and 2; orthogonal offset 2 considers the average speed of subsystems 2 and 3; orthogonal offset 3 considers the average speed of subsystems 3 and 4; and each internal offset considers the average speed of its own subsystem. Note that the average speed of a subsystem means the average speed of the vehicles on all of the links that are directed towards the intersections within the subsystem. As a result of this set-up, there is some degree of cooperativeness between each orthogonal offset and the corresponding internal offsets.

The NES parameters are set up as follows. Dither amplitude \( a = 3 \); \( k = 2.0 \) and \( k = 1.0 \) for the orthogonal and internal offsets respectively; \( \omega = 10^{-3} \); \( \omega_I = 1 \times 10^{-4} \text{ rad/s} \); \( \omega_L = 5 \times 10^{-3} \text{ rad/s} \); \( \omega_{\text{ort}} = \omega_{\text{int}} = 0.88 \); \( \omega_{\text{ort2}} = 0.82 \); \( \omega_{\text{ort1}} = \omega_{\text{int3}} = 1.0 \), and \( \omega_{\text{ort2}} = \omega_{\text{int4}} = 0.94 \), where the subscript ort and int indicate orthogonal and internal offset respectively. Some of the dithers are re-used since each offset only significantly affects the performance measure of the “adjacent offsets/agents”. Lastly, \( \phi = 35^\circ \) and \( \phi = 23^\circ \) for the orthogonal and internal offset respectively.

Fig. 9. (SCATS) The average speed of the east bound traffic approaching the nodes within subsystem 2 for various values of the internal offset of subsystem 2. The traffic demand is East bound biased Case 1. The average speed of all other directions and all other nodes are unaffected (a flat plot similar to Node 5).
In the simulation, the offsets are initialised based on the approximated travel time between intersections. With an approximate average speed of 10 m/s (from Fig. 10) and link length of 300 m, the travel time, and hence the initial offset, is approximated to be 30 s (for both internal and external offsets). The simulation result is shown in Figs. 11 and 12. It can be observed that the optimal offsets for Case 1 of both internal and orthogonal offsets are approximately 15 s. In addition, it can be observed that the output corresponding to each offset is improved. On the other hand, the optimal offset for Case 2 is higher compared to Case 1 due to the slower speed. Also, the improvement in the output corresponding to each offset is predictably smaller because the initial offsets are closer to their optima.

It has been demonstrated that augmenting the existing SCATS algorithm with a Nash extremum seeker operating on the offset parameters has enabled the optimal average network speed to be obtained in reasonable time. Other tuning parameters and alternative measures of network performance satisfying the general requirements of NES could be substituted without compromising the ability of the augmented system to achieve optimal performance. Furthermore, similar results will be obtained if SCATS were replaced by another non-model based alternative.
6. Simulation results for a non-model based strategy augmented with NES

In this study, an example of a traffic light control strategy from the non-model based category that is investigated is SOTL (de Gier et al., 2011; Zhang et al., 2013). This version of SOTL is a highly adaptive signal system that has been shown in simulations to perform as well as, and often better than, SCATS (Zhang et al., 2013). In this section, it is shown that SSIO maps for SOTL exist and the NES scheme is able to seek the optimal parameters. In addition, it is also shown that NES can be implemented to deal with a time-varying traffic scenarios, depicting periodic day-to-day traffic fluctuations.

6.1. Steady-state input–output map

In SOTL, each node has its own threshold value. Interestingly, when SOTL is used in the network studied, it appears that each node’s output is independent of other nodes’ inputs. When the threshold of each node is perturbed by sinusoid signals of different frequencies, Fig. 13 shows that the spike in each node’s output (average speed) power spectrum curve occurs predominantly at the frequency corresponding to its own input (threshold). In conclusion, each node is affected most significantly by its own input. Thus, when SOTL is used, the NES scheme has 16 parameters and the problem can be treated as multiple SISO optimisations, which implies that unrestricted dither re-use is allowed. Hence, only one value of dither frequency is going to be used for SOTL.
The SSIO maps of Node 6 (which is typical of each node), for various values of $\alpha$ and $\beta$, when using SOTL is shown in Fig. 14. Firstly, it is found that when $\alpha/C_2 > 11$, the traffic in the network has an average speed of approximately 40 km/h (compared to the average free-flow speed of 60 km/h), whereas $\alpha/P_0 > 18$ corresponds to a congested network where the speed drops to below 15 km/h. Secondly, it is found that when $\alpha$ is not very high, $\beta$ does not affect the speed in the network unless it is very low, as shown in Fig. 14b. Most importantly, it can be observed that the optimal threshold value varies for different values of $\alpha$. Note that in this case, it is sufficient to demonstrate the usefulness of the NES scheme when adapting SOTL without having to introduce internal sources and sinks.

6.2. Time-invariant traffic demand

In this section, the performance of the NES scheme for adapting the threshold value of SOTL by using constant traffic demand profiles is to be investigated. As before, the purpose of this section is to demonstrate the convergence of the NES scheme to the various optima when faced with different traffic conditions. Furthermore, the NES parameters are set up as follows: dither amplitude $a = 1$; $k = 0.4$; $\omega = 10^{-3}$; $\omega_H = 9 \times 10^{-5}$ rad/s; $\omega_L = 3 \times 10^{-3}$ rad/s; and $\phi = 60^\circ$. Note that each node is using the same dither frequency.

There are two cases considered here, where the first simulates a medium-load demand (such as off-peak periods) and the second approximates a heavy traffic condition. The first case considered is a uniform traffic demand with $\alpha = 0.13$ and $\beta = 0.9$. As can be observed from Fig. 15, the NES scheme is able to adapt the threshold values to an equilibrium value of approximately 4.5 and the average speed is improved, especially the case with initial conditions equal to 10. In addition,
shown in Fig. 16 is a typical convergence trajectory of the NES scheme with the second traffic demand case, which is uniform with $a = 0.17$ and $b = 0.9$. It can be observed that the NES scheme again converges to the optimum, and the threshold is different to the first case.

6.3. Time-varying traffic demand

To demonstrate the usefulness of the proposed NES scheme when using SOTL on a daily basis (as mentioned in Section 2.4), time-varying demand profile, as shown in Table 6, is used in the simulation. Although in this case the demand is uniform from all direction, it is sufficient to show that the NES scheme is able to seek the optima for all three periods: night time, peak and off-peak.

Table 6  
SOTL traffic demand cycle for each day.

<table>
<thead>
<tr>
<th>Duration (h)</th>
<th>Night time</th>
<th>4.5 Morning peak</th>
<th>6 Off-peak</th>
<th>3.5 Afternoon peak</th>
<th>3.5 Evening off-peak</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>0.07</td>
<td>0.16</td>
<td>0.12</td>
<td>0.16</td>
<td>0.12</td>
</tr>
<tr>
<td>$b$</td>
<td>0.99</td>
<td>0.70</td>
<td>0.90</td>
<td>0.70</td>
<td>0.90</td>
</tr>
</tbody>
</table>

Fig. 15. (SOTL) The simulation result of Node 6, which is typical of all nodes, using two initial threshold values when $a = 0.13$ and $b = 0.9$.

Fig. 16. (SOTL) The simulation result of Node 6, which is typical of all nodes, using two initial threshold values when $a = 0.17$ and $b = 0.9$. 

Fig. 17 shows the typical convergence of the SOTL threshold under the traffic demand profile in Table 6. It can be observed that for each period, the NES scheme converges to the vicinity of its corresponding optimum (approximately at 2, 4, and 6 for night time, off-peak and peak respectively). Furthermore, Fig. 18 shows the improvement in the average speed of each node.
Fig. 17. (SOTL) The convergence of the threshold of Node 6, which is typical of all nodes, during the night (black), off-peak (red), and peak (green) period. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Fig. 18. (SOTL) The average speed of each node for each day. The top group of lines represents the corner nodes (Nodes 1, 4, 13, and 16); the middle group of lines represent the edge nodes (Nodes 2, 3, 5, 8, 9, 12, 14, and 15); and the bottom group of lines represent the internal nodes (Nodes 6, 7, 10, and 11).
during each day. Except during peak period, the improvement in the average speed is approximately up to 6.9%, 6%, and 5.4% during the night time, off-peak, and whole-day period respectively. Note that for the peak period, the average speed does not improve significantly. The reasons for this are twofold. Firstly, the threshold has been initialised sufficiently close to the optimum. Secondly, as previously shown in Fig. 14a, when $\alpha = 0.16$, the average speed is almost the same for threshold values from 6 to 8. Thus, the average speed during the peak period is already close to the maximum. This also explains why the threshold value chosen by the NES scheme varies quite a lot during peak when compared to other time periods.

In conclusion, the NES scheme is able to adapt the threshold of the SOTL controller to a value that optimises the average speed at each node, for each traffic condition. By using the NES scheme, it is also possible to calibrate other tuning parameters of SOTL that optimise a different performance measure, as long as they satisfy the general properties required by the NES scheme.

7. Conclusions and future work

Using ES as an augmentation to existing urban traffic control strategies has been demonstrated to improve their performance in a uni-modal urban environment. This improvement is achieved by real-time fine tuning of their parameters and has been demonstrated for three different traffic light controllers, namely: perimeter control, SCATS, and SOTL. It has been shown in each case that the proposed NES scheme is able to seek the optimal parameters with respect to a range of different performance measures, irrespective of the type of the traffic light control strategy. This clearly demonstrates the versatility of the NES scheme for performance optimisation without requiring an overhaul of the existing traffic light controllers. In addition, one of the most important benefits of using the proposed NES scheme comes from the fact that the stability has been theoretically guaranteed (Kutadinata et al., 2015, in preparation). Consequently, the NES scheme can always be fine-tuned such that it is able to seek the optimal parameters of the traffic controller.

The application of the NES scheme for urban traffic light control can be extended to include more complex and realistic scenarios. This includes: adaptation of more parameters; simulating more realistic networks; consideration of multi-modal traffic; accommodating flow prioritisation; estimation of performance metrics that cannot be directly measured; and more thorough investigation about other adaptation strategies followed by a comparison study with the developed NES scheme.

Acknowledgements

This work was supported under the Australian Research Councils Linkage Projects funding scheme (project number LP120100258), and T.G. and C.M. are the recipients of Australian Research Council Future Fellowships (project numbers FT100100494 and FT100100538). This research was undertaken with the assistance of resources provided at the NCI National Facility through the National Computational Merit Allocation Scheme supported by the Australian Government.

Appendix A. Details of the perimeter control

The perimeter control used in this paper is based on the work by Aboudolas and Geroliminis (2013) and Ampountolas et al. (2014). In this strategy, the number of vehicles inside the network, $n$, is assumed to be governed by the following dynamics:

$$\dot{n} = \theta Q_{in} - Q_{out}(n), \tag{A.1}$$

where $Q_{in}$ and $Q_{out}$ is the flow of the incoming and outgoing vehicles respectively, and $\theta \in [0, 1]$ is the proportion of $Q_{in}$ that is actually allowed to enter the network. Note that $Q_{in}$ is assumed to be constant in (A.1), unlike its counterpart in the work by Aboudolas and Geroliminis (2013) where $Q_{in}$ is proportional or analogous to $Q_{out}(n)$. Although $n$ is not normally equivalent to the density inside the network, since all links in the network under consideration (by the perimeter control) are identical, we can treat $n$ and the density as being equivalent in this study. Furthermore, $Q_{out}(n)$ represents the MFD of the network if input to output dynamics are not instantaneous and any delays are comparable with the average travel time across the network.

It is desired to regulate the system to a set point $(n^*, \theta^*)$, around which the system is to be linearised. The set point $n^*$ should be selected as close as possible to the critical density, whereas $\theta^*$ needs to be determined by solving $\theta = \theta^* Q_{in} - Q_{out}(n^*)$, which yields $\theta^* = Q_{out}(n^*) / Q_{in}$. Note that $\theta^* < 1$ only when $Q_{in} > Q_{out}(n^*)$. This implies that the perimeter control is only applicable when the traffic demand is sufficiently high. Furthermore, by defining the error coordinates $\tilde{n} = n - n^*$, $\tilde{\theta} = \theta - \theta^*$ and using first-order Taylor approximation, the linearised system can be written as follows:

$$\dot{\tilde{n}} = -\frac{dQ(n^*)}{dn} \tilde{n} + Q_{in} \tilde{\theta}.$$ 

Moreover, in order to remove the requirement of the knowledge of $\theta^*$, an LQI regulator is used. Thus, the system is augmented by the following error variable:

$$\dot{\tilde{\theta}} = \tilde{n}.$$
Then, the system is discretized and a state-feedback control law of the following form can be derived:

$$\theta(t_d) = \theta((t_d - 1) - k_p(n(t_d) - n(t_d - 1))) - k_i(n(t_d) - n^*)$$

(A.2)

where \(t_d\) is the discrete time index, \(k_p\) and \(k_i\) are the controller gains. Aboudolas and Geroliminis (2013) outlines the derivation of (A.2) in detail.

References


